MC-OLA：基于马尔可夫链的多表连接在线聚集技术 *

史英杰 1，杜　方 2

(1. 北京服装学院 信息工程学院, 北京 100029; 2. 宁夏大学 信息工程学院, 银川 750021)

摘　要：多表连接是大数据分析领域重要的查询类型之一，然而连接查询的实现代价很高，从而影响了大数据分析结果的时效性。在线聚集能够在查询完成前反馈具有统计意义的估计结果，具有重要的意义。目前已有的多表连接在线聚集算法从各表进行统一随机采样，导致连接结果的产出率低，并且导致分组连接查询的估计准确率低。针对这一问题，提出了基于马尔可夫链的多表连接在线聚集技术，将多表连接的实现过程转换为马尔可夫链上的随机游走过程，确定好连接顺序后在游走起始层创建分层样本，并设计了相应的采样策略及结果估计方法。将所提出技术在在线化 Hadoop 平台上实现，实验结果证明所提出方案的响应时间优于已有算法，并且具有良好的扩展性。

关键词：在线聚集；马尔可夫链；分层采样；多表连接

中图分类号：TP391　　doi: 10.3969/j.issn.1001-3695.2018.07.0384

MC-OLA：multi-table join online aggregation based on Markov chain

Shi Yingjie[1], Du Fang[2]

(1. *School of Information Engineering*, *Beijing Institute of Fashion Technology*, *Beijing* 100029, *China*; 2. *School of Information Engineering Ningxia University* 750021, *China*)

Abstract: Multi-table join is one of the most important query operations in the field of big data analysis, however, the implementation of multi-table join is expensive, which affects the timeliness of the big data analysis results. Online aggregation provides feedback of statistical significance far before the query finishes, which is of great significance. The existing work on multi-table join online aggregation conducted uniform sampling on every joining table, which results in low join result yield and estimation inaccuracy on grouping join queries. To address this problem, this paper proposed the multi-table join online aggregation technique based on Markov chain, transformed the multi-table join process into the random walk on Markov chain, constructed stratified sample on the walk start strata after determining the join order, and designed the corresponding sampling mechanism and estimation algorithm. The experiment was conducted on the online Hadoop platform, and the results demonstrate that the response time of technique proposed in this paper outperforms the existing algorithms, and it owns efficient scalability.

Key words: online aggregation; Markov chain; stratified sampling; multi-table join

0　引言

社交媒体、移动设备及传感器以前所未有的速度持续产生着海量数据，探索这些数据背后蕴藏的价值已经成为目前工业界及学术界十分关注的问题，然而复杂的数据分析任务在海量数据上运行缓慢，使得分析结果的时效性和价值大打折扣，成为数据驱动任务发挥作用的瓶颈。即席交互式数据分析在决策支持、趋势分析及数据可视化等领域发挥重要的作用，成为目前大数据分析领域亟待解决的问题之一。在线聚集不断对部分样本数据进行处理，从而可以在较短时间内返回具有统计意义的估计结果，为即席交互式数据分析提供了一种全新的解决方案。在线聚集于20世纪90年代在关系数据库领域被首次提出，随后取得了一系列的研究成果，然而在关系数据库市场所带来的影响力十分有限。随着大数据与云计算平台的出现，新型的数据模式和数据管理方式为在线聚集带来了发展机会。然而目前在云计算平台的在线聚集研究大多关注单表上的操作，或者简单的两表连接，针对多表连接的研究工作还比较少。多表连接是决策支持、数据挖掘和分析中最重要的操作之一，在大数据决策支持应用的基准测试 TPC-H[1]中，22 条查询语句中的 17 条是连接查询，最多涉及 8 表的连接。

相对于单表或两表连接在线聚集，针对多表连接的在线聚集处理方式更加复杂，已有的工作无法直接应用。a)多表连接

收稿日期：2018-07-22；**修回日期：**2018-09-07　**基金项目：**国家自然科学基金资助项目（61502279）；北京市教委科技计划项目（KM201710012008）；北京服装学院高水平教师队伍建设专项资金资助项目（BIFTQG201803）

作者简介：史英杰（1983-），女，山东滨州人，副教授，博士，主要研究方向为云数据管理，大数据在线聚集，大数据系统的性能分析与优化（shiyingjie1983@163.com）；杜方（1974-），女，教授，博士，主要研究方向为智能信息检索、大数据管理.

类型呈多样化，包括链式连接、非环型连接、环型连接等，不同连接类型的在线查询处理方法及结果估计方法均不相同；b)多表连接的结果空间随着连接表数的增大呈指数级增长，而选择率通常较低，已有的采样方法将导致多表连接的结果产出率极低；c)多表连接总体的数据分布不是由一个表简单决定，而是多个表相互影响的结果，已有的解决小分组等问题的算法无法应用。针对上述问题，本文提出了基于马尔可夫链的多表连接在线聚集技术 MC-OLA，将多表连接处理过程转换为马尔可夫链上的遍历游走过程，基于该模型在游走起点创建分层样本，并针对采样方法进行结果无偏估计和置信区间计算。本文将 MC-OLA 在在线化 Hadoop 平台 HOP[2]上实现，实验结果表明 MC-OLA 的性能优于 random walk 算法及 ripple join 算法，且具有良好的数据扩展性。

1 相关工作

在线聚集于 20 世纪 90 年代在关系数据库领域被首次提出[3]，其主要思想是在查询处理过程中不断从总体数据进行采样，基于样本对查询结果和置信区间进行估计，随着处理数据的不断增长，查询结果的估计质量不断提升。针对单表查询的在线聚集操作，文献[4]给出了无偏估计和置信区间的具体计算方法。随着大数据技术的发展，在线聚集技术在云计算领域又引起了广泛关注。文献[5]基于 MapReduce 框架提出了一种采用贝叶斯理论实现在线聚集的方法，考虑每个数据块的聚集值与该数据块处理时间的关系，将数据块的聚集值、调度时间以及处理时间一起进行统计建模。该方法假设数据块的处理时间越长，那么其聚集值也越大，该假设并不是在所有的聚集操作中均成立，而且实现方法也比较复杂。G-OLA[6]基于 Spark 平台实现，主要面向嵌套查询估计的优化，G-OLA 采用统一随机采样技术进行结果估计，利用嵌套查询估计结果及置信区间，将最终查询结果分成确定和不确定两个部分，从而减少更新过程中的数据操作。BlinkDB[7]同样基于 Spark 平台实现，分析查询负载并基于查询语义选择分层采样的列集，利用分层采样技术解决小数据分组问题，但未解决低选择率的问题。PF-OLA[8]提出了一个在分布式环境中实现在线聚集的框架，包括节点内部数据存储、分布式数据采样、查询处理以及结果估计实现机制，使得分布式环境对在线聚集数据流的影响降到最低。文献[9]提出了一种面向原始数据的双层次采样算法，在数据处理过程中分别从数据块及数据块内部进行采样，该算法能够根据计算资源实时调整采样大小，从而减少采样代价。CrowdOLA[10]将基于众包的实体解析方法与在线聚集技术相结合，解决了在重复数据上进行结果估计准确性不高的问题。总的来说，上述研究工作主要关注单表查询的在线聚集实现技术。

单表查询在线聚集的实现机制无法直接应用于多表连接查询的在线聚集，因为从各个表进行随机采样后连接的结果具有相关性，并不是完全随机的。Haas 等人[11]对该问题进行研究，并提出了 ripple join 算法。Ripple join 从各连接表中轮流随机采样，并将样本数据放入内存。每当新的样本从其中一张表中读取出来，将会和其他表中已经读取到的所有数据进行连接，该过程反复执行，直至估计结果满足用户需求时停止。因为从各个表中抽取样本数据时并不考虑数据分布、查询负载等信息，所以当满足连接谓词的结果较少或者分组较多时，ripple join 的估计结果产出率非常低。针对基本 ripple join 算法的不足，随后关系数据库领域出现了对其扩展的研究工作。文献[12]将 ripple join 算法进行了并行化处理，然而该方法并不具有扩展性，一旦内存无法再加载数据时，估计结果将不具有统计意义；文献[13]将 sort-merge 思想应用到 ripple join 算法中，对内存换出到外存的数据进行随机化处理，从而保证估计结果的统计意义，并在引擎 DBO 上实现；文献[14]提出了 TurboDBO，对查询处理过程中的中间结果进行有效利用，从而进一步加快置信区间的收敛速度。COLA[15]基于 MapReduce 框架实现了两表连接在线聚集，考虑到云计算平台中数据存储的特点，采用基于数据块的双重分层采样方法进行数据抽样，对 ripple join 算法进行并行化，并设计 MapReduce 作业进行实现。总的来说，ripple join 盲目的从各连接表中随机抽取数据，当连接谓词的选择率较低、或者连接结果的分组较多时，连接结果的产出率非常低，从而导致置信区间的收敛速度缓慢。文献[16]提出了 wander join 算法，在连接数据表上进行随机游走，在游走过程中利用连接列上的索引确定游走方向，基于每次游走对结果进行估计。Wander join 解决了 ripple join 在连接选择率较低时，估计结果产出率低的问题，然而当分组较多或数据倾斜出现时，仍然会出现置信区间收敛速度缓慢，甚至会出现小分组估计结果丢失的情况。本文使用马尔可夫链对多表连接过程进行建模，并基于分组列集进行分层采样，有效应对连接负载或数据倾斜导致的结果估计不准确及置信区间收敛速度缓慢的问题。

2 MC-OLA 概述

2.1 问题建模

采用四个表的自然连接说明 MC-OLA 的建模过程，设连接形式为

$$\text{SELECT } op(exp(t_{1i}, t_{2j}, \ldots, t_{km})) \text{ FROM } R_1, R_2, R_3, R_4$$
$$\text{WHERE } R_1.A = R_2.B \text{ and } R_2.C = R_3.\text{D and } R_3.\text{E}=R_4.\text{F} \quad \text{GROUP BY } col;$$

在上述查询表达式中，*op* 是具体的聚集操作，*exp* 是对元组的代数操作，*col* 是分组列集。假设连接的顺序是 R1-R2-R3-R4，将连接过程转换为从 R1 到 R4 的马尔可夫随机过程，如图 1 所示。图中节点表示各表中的元组，若两个节点满足连接谓词，则在两点间存在一条边。例如，t21 与 t31 间存在一条边，表示满足条件 t21.C=t31.D。从 t21 还有可能游走到 t32 和 t35，然而这种选择游走方向的概率和 t21 之前的路径无关，因此满足马尔可夫性质。从 R1 中某一元组开始游走到 R4 某一元组结束所形成的随机路径，即为一个连接结果。

2.2 系统架构

MC-OLA 的系统架构由样本创建和在线聚集两个阶段构

成，如图 2 所示。样本创建阶段结合负载特征为原始数据集创建分层样本，分层依据为查询负载中的分组列集。分组列集选择采用文献[17]提出的方法，使得列集在负载中出现的概率以及负载中的分组列集被覆盖的概率达到最大。基于确定好的分组列集以及索引的分布情况，MC-OLA 确定各表的连接顺序，进而在马尔可夫链的游走起点创建分层样本。在在线聚集阶段，对用户提交的多表连接查询语句进行解析，动态选择查询代价最小的样本进行分层采样，进而估计查询结果及置信区间。

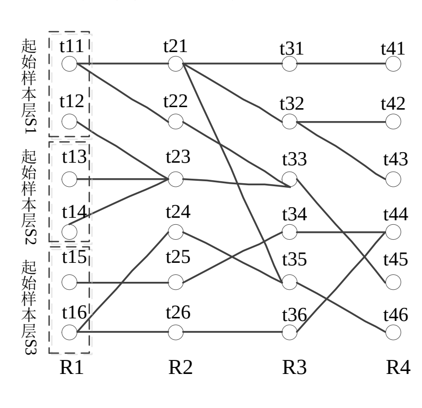


图 1　四表链式连接图

3　采样技术

单表在线聚集的估计总体为表中的全部元组，全表扫描即可获得各层样本及样本层大小。相对于单表查询，多表连接的估计总体为连接结果或笛卡尔积的子集，其分层样本创建要复杂得多。若采用传统的 ripple join 算法，其总体为各表笛卡尔乘积的子集，计算复杂度随连接表数的增加呈指数级增长。MC-OLA 将连接过程在马尔可夫链上进行建模，总体被看做从起始表元组到终点表元组的随机游走路线，分层样本的创建通过遍历马尔可夫链实现。

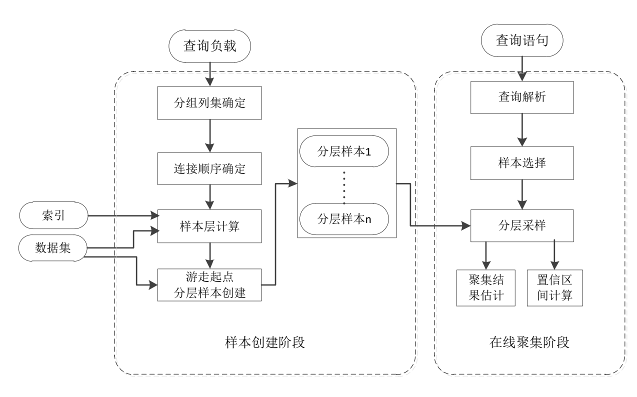
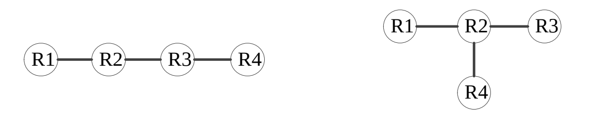


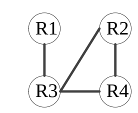
图 2　MC-OLA 系统架构图

3.1　确定连接顺序

2.1 节的问题建模基于链式连接介绍，而多表连接还包括非环型连接和环型连接。用节点表示连接表，节点间的边表示两表间存在连接关系，则四表的连接类型如图 3 所示。给定一个多表连接的查询，其实现连接的顺序有很多种，而不同的连接顺序对采样和结果估计的准确性及收敛速度产生不同的影响。因此在创建分层样本之前，MC-OLA 首先根据负载特征和索引分布情况确定连接顺序。以图 3(a)中的链式连接为例，R_1-R_2-R_3-R_4 以及 R_3-R_4-R_2-R_1 均是合理的连接顺序，而 R_3-R_1-R_2-R_4 则不是正确的连接顺序。引理 1 给出了多表连接顺序的确定准则。



(a) 链式连接　　　　(b) 非环型连接



(c) 环型连接

图 3　四表连接类型

引理 1　设查询语句中参与连接的表有 m 个，则连接顺序 R_1-R_2-R_3…R_m 为合理连接顺序的充分必要条件为：对于连接顺序中的任意表 R_i，排在 R_i 前面的表格中至少有一个与 R_i 有直接连接关系。

证明　采用数学归纳法进行证明。

a）当有两个表 R_1 与 R_2 进行连接时，连接顺序包括 R_1-R_2 或者 R_2-R_1 两种，显然满足条件。

b）假设有 k 个表进行连接时，命题成立。

充分性。设 k 个表的连接序列为 R_1-R_2-R_3…R_k，且满足“排在 R_i 前面的表格中至少有一个与 R_i 有直接的连接关系”条件，则当增加一个表 R_{k+1} 参与连接时，将 R_{k+1} 表放置在原连接序列中 R_i 和 R_{i+1} 之间，且满足 R_1-R_i 的表中至少有一个与 R_{k+1} 有直接连接关系，则从 R_1 到 R_{k+1} 可完成连接，连接后的结果与 R_{i+1} 到 R_k 的序列也可完成连接，因此连接顺序合理。

必要性。设 k 个表的连接顺序为 R_1-R_2-R_3…R_k，且满足“排在 R_i 前面的表格中至少有一个与 R_i 有直接的连接关系”条件，则当增加一个表 R_{k+1} 参与连接时，将 R_{k+1} 表放置在原连接序列中 R_i 和 R_{i+1} 之间，且新的序列是合理的连接序列。则 R_1 到 R_i 的序列中必定至少有一张表与 R_{k+1} 存在直接连接关系，因此新的合理连接序列仍然满足直接连接关系的条件。证明完毕。

基于引理 1 给出连接顺序确定算法。首先根据索引情况为连接图添加方向，若 R_i 与 R_j 间存在一条连接边，且 R_j 在连接列上有索引，则添加方向为 R_i 到 R_j，反之亦然。设分组列集属于表 R_i，接下来从 R_i 开始对有向图进行顶点遍历生成连接序列，如算法 3.1 所示。该算法产生的连接序列为连接图的生成树，对于环型连接，所生成的连接序列没有包含全部连接关系。可在游走完成后，利用剩余的连接关系对游走的连接结果进行进一步筛选。例如，对图 3(c)中的查询，若分组列集位于表 R_3，且生成的连接序列为：R_3-R_1-R_2-R_4，利用 R_2-R_3 的连接关系在游走完成后对连接结果进行筛选。

算法 1　连接顺序确定算法

Input: *JoinArray[N][N]*: two-dimensional array containing the directed join query graph;

Output: *JoinList*: join order list;

1: JoinSearch(**int** *n*, **int** *m*)

```
2: {   JoinList.add(n);
3:   for(int i=0;i++;i<m)
4:     if(JoinArray[n][i]==1&visited[i]==0)
5:     {  visited[i]=1;
6:        JoinSearch(i,m);
7:     }
8: }
```

3.2 创建分层样本游走起始层

MC-OLA 将分组列集所在表放置在马尔可夫链随机游走的起始端，基于起始端的数据表 Rs 创建分层样本的游走起始层。针对单表查询创建分层样本时，样本总体即为原始表数据，因此直接根据分组列集进行分层，各样本层中元组的个数即为层大小。在多表连接的在线聚集中，样本总体为多表连接的结果，无法通过单独扫描任何一个表格得到分层样本。针对 Rs 中的任一元组 ti，MS-OLA 基于马尔可夫链从 ti 进行游走，计算与该元组相关联的连接结果的个数，从而进一步确定 ti 起始层的大小。

以图 1 的四表链式连接为例对样本创建过程进行介绍，若连接序列为 R_2-R_1-R_3-R_4，则以 R_2 为起始端进行游走，并创建分层样本的游走起始层。假设游走的起始元组为 t21，当游走至 R_1 中的元组 t11 时，无法继续前进，因此跳转到 t21 后向 R_3 方向继续游走，直至游走到 R_4 中的元组 t41。MC-OLA 将连接图中度为 1 的表定义为“边缘表”，包括 R1 这类游走过程中遇到后需要跳转方向的表，以及 R4 这类标志着游走结束的表。一旦游走过程中遇到“边缘表”，MC-OLA 记录当前游走分支的路径条数并改变游走方向，最终的连接结果数为各个分支路径条数的乘积，具体实现过程如算法 3.2 所示。给定游走起始元组 *t*，沿着 *t* 所在表的所有邻接表开始游走(第 4 行至第 15 行)。若连接的邻接表 R’ 在连接图中的连接度小于 2，则说明 R’ 是“边缘表”，调用算法 getPathNum 计算分支路径的条数（第 8 行至第 9 行）；否则，说明 R’ 还能沿着连接序列继续游走，递归调用算法 getJoinSize 获取连接结果条数(第 10 行至第 13 行)。最终将各个分支的连接结果相乘，得到以 *t* 为游走起始点的连接结果数。游走分支路径条数确定方法如算法 3.3 所示，给定分支路径的起始元组 *t* 及游走方向上的邻接表 R，根据 R 在连接列上的索引获取与 t 相连接的元组（第 4 行），对元组数进行累加获得分支路径的条数（第 5 行至第 9 行）。

令 $V=\{V_1, V_2, \cdots, V_L\}$表示数据集在分组列集上的不同取值，可将游走起始表分成 *L* 个不同的分区，每个分区即分层样本起始层的“一层”，每层的大小由以该层中所有元组开始游走的路径数决定。MC-OLA 扫描分组列集所在表的元组，并使用 getJoinSize 算法计算各元组为游走起始点的连接结果数，从而创建分层样本起始层。尽管创建样本需要扫描连接起始表，并且需要在多个表中进行游走，然而顺序扫描起始表数据的吞吐率要远远高于随机读取数据的吞吐率，并且沿着多表马尔可夫链的游走依据索引进行，因此创建样本的代价在可接受范围之内。

算法 2 连接结果大小确定算法

Input: tuple *t*: walk start tuple; *R*: table containing *t*; *L*: join order list;

Output: int *joinSize*: number of joined results with *t* as the walk start tuple；

```
1: int getJoinSize(tuple t)
2: {  int joinSize=1;
3：  int tempSize;
4：  for(int i=0; i++; i<join degree of R)
5:    {  R’= adjacentTable(i);
6：       if(R’ is visited)
7：          continue;
8:        if(R’ join degree < 2)
9:          tempSize=getPathNum(t, R’);
10:        else
11:        {  while(t’=walktowards (R’))
12:             tempSize +=getJoinSize(t’);
13:        }
14:        joinSize*=tempSize;
15:    }
16:  return joinSize;
17: }
```

4 在线聚集实现

4.1 采样策略

目前实现连接在线聚集的主流算法是 ripple join，ripple join 直接从各个连接表中分别进行统一随机采样，并不考虑数据分布对连接结果的影响。因此当出现连接选择率较低的情况时，会导致结果产出率低及小分组的估计准确率低。MC-OLA 采用马尔可夫链对连接进行建模，将连接过程转换成多表上的随机游走，从游走起始端的数据进行分层采样。在线聚集以固定的频率更新结果，依据更新频率可计算出每次抽取的样本大小 *N*。MC-OLA 并不是将各表的原始数据作为总体，而是将连接结果作为总体，因此 *N* 表示连接结果的大小，即随机游走路径的条数。在采样过程中，将 *N* 分配至各个分组的样本层。为了使估计结果的方差和最小，MC-OLA 的样本大小分配算法参考文献[18]提出的类平均算法。从每个分层中抽取样本的大小为 *N* 在 *L* 个样本层中的平均值和剩余样本的最小值，若总的样本数量小于 *N*，则将 *N* 扩大至 *N’*后重复上述过程，直至找到使得总采样数量最接近 *N* 的 *N’*。与文献[18]中的算法不同，MC-OLA 各层样本的剩余值是指连接结果数，并非游走起始表的元组数。

算法 3 游走分支路径条数确定算法

Input: tuple *t*: walk start tuple; table *R*: the adjacent table according to the walk direction

Output: int *pathNum*: number of walk path with *t* as the walk start tuple；

```
1:   int getPathNum( tuple t, table R)
2: {
```

```
int pathNum=0;
tuple t'=walkAccordingtoIndex(t,R);
while(t')
{
  pathNum++;
  t'=walkAccordingtoIndex(t,R);
}
return pathNum;
}
```

4.2　聚集结果及置信区间估计

确定好每层的采样数量后，从样本起始层开始进行随机游走，游走的次数为该层所分配的样本数。查询语句基于 2.1 节问题建模的连接形式，聚集操作主要讨论 SUM 和 COUNT 的实现方法，其他的聚集操作如 AVG、STD-DEV 等可通过相应的扩展实现。采用图 1 所示的链式连接为例介绍各层连接结果的游走方法，图 1 中的起始数据被分为三个样本层，聚集结果及置信区间的估计在每个分组内分别进行。当对样本层 *S1* 进行随机游走时，首先从 *S1* 中随机等概率抽取一个起始元组。假设 t11 被抽取到，接下来根据 R2 在与 R1 连接键上的索引，从 R2 中与 t11 相邻的元组中随机抽取元组，依次沿着马尔可夫链继续向下游走，直至游走到 R4 结束，最终抽取出一条路径。根据多次游走抽取出的连接结果，对聚集结果和置信区间进行估计。在该例中样本层 S1 里含有 6 条路径，若抽取的路径为 t11-t21-t32-t42，则被抽取到的概率为 1/24，并非 1/6。利用该方法抽取每条路径的概率不相同，所以连接结果的样本并不是统一随机抽样获得。

设起始层样本分别为 $S_1, S_2, \ldots, S_m$，给定样本 S_i，该层中每条路径 λ 被抽取的概率为

$$P(\lambda)=\frac{1}{|S_i|}*\frac{1}{B_2(t_1)}*\frac{1}{B_3(t_2)}*\ldots*\frac{1}{B_N(t_{N-1})}$$

其中: $B_2(t_1)$ 为 R2 中与 R1 的元组 t1 满足连接关系的元组。设 $op(\exp(\lambda_j))$ 为路径 λ_j 所对应的连接结果上的聚集

操作，定义随机变量 $\exp_p(\lambda_j)$ 的取值为若 op 为 SUM 操作，$\exp_p(\lambda_j)=\exp(\lambda_j)$；若 op 为 COUNT 操作，$\exp_p(\lambda_j)=1$。给定估计的分组，设该分组样本大小为 n，定理 1 给出了聚集结果及置信区间的估计方法。

定理 1　$\tilde{\mu}=\frac{1}{n}\sum_{j=1}^{n}\frac{1}{P(\lambda_j)}*\exp_p(\lambda_j)$ 是对多表连接聚集结果的无偏估计，设估计结果的置信度为 ρ，则置信区间为 $I=[\tilde{\mu}-z_\rho\tilde{\sigma}_n/\sqrt{n},\tilde{\mu}+z_\rho\tilde{\sigma}_n/\sqrt{n}]$，其中 $\tilde{\sigma}_n^2=\frac{1}{n-1}\sum_{j=1}^{n}[\frac{1}{P(\lambda_j)}*\exp_p(\lambda_j)-\tilde{\mu}]$。

证明　路径 λi 被抽取的概率为 $P(\lambda_i)$，每条路径被抽取的概率不同，属于独立的有偏采样。根据 Horvitz–Thompson 有偏采样估计原理，可知 $\frac{1}{P(\lambda_j)}*\exp_p(\lambda_j)$ 是对总体聚集值的无偏估计[19]，而且 n 个游走路径是独立的，所以其均值仍然是对总体聚集值的无偏估计。构建随机变量 $X_j=\frac{1}{P(\lambda_j)}*\exp_p(\lambda_j)$，且满足 $\sum_{j=1}^{n}P(\lambda_j)=1$，则对连接结果聚集值的估计转换为对新变量总体均值的估计，根据中心极限定理，可得出总体均值近似服从正态分布：$\tilde{\mu}\sim Normal(\mu,\sigma^2/n)$，对该正态分布进行标准化处理后可得 $Z=(\mu_{blk}-\mu)/(\frac{\sigma}{\sqrt{n}})\sim Normal(0,1)$

给定置信度 ρ，可得 $P\{-z_\rho\le Z\le z_\rho\}=\rho$，使用样本方差 $\tilde{\sigma}_n^2$ 代替总体方差 σ^2，即可得 $P\{\mu-z_p\tilde{\sigma}_n/\sqrt{n}\le\mu\le\mu+z_p\tilde{\sigma}_n/\sqrt{n}\}=\rho$。证明完毕。

5　实验结果分析

本文基于 HOP[2]平台的 MapReduce 框架分别实现了 MC-OLA、random walk 和 ripple join，从估计结果的准确性和置信区间的收敛速度两个方面进行性能测试和比较分析，并对数据集大小的扩展性进行了测试。Random walk 采用 MC-OLA 的连接顺序确定方法，不同的是，random walk 从起始连接表中进行统一随机采样，然后基于样本利用连接列上的索引进行游走，直至游走到连接序列中的最后一张表。实验采用大数据决策支持测试基准 TPC-H[1]中的查询负载及数据集进行。

5.1　实验配置

实验测试平台是由 11 个节点构成的集群环境，节点间通过 1 Gbit 的交换机相连，其中一个节点作为 HDFS 和 MapReduce 的主节点，剩余的 10 个节点作为工作节点，软件运行平台为 HOP。每个节点拥有 4GB 内存和 3.3GHZ 的四核 CPU，每个节点的磁盘大小为 1TB。设置文件系统 HDFS 的块大小为 64M，并在每个节点上配置 2 个 map 任务和 1 个 reduce 任务。

实验的连接查询负载采用 TPC-H 中的 Q5、Q11 和 Q21。其中 Q5 是六表非环型连接，Q11 是三表链式连接，Q21 是四表链式连接。涉及的表格包括：customer、orders、lineitem、supplier、nation、region。为了着重分析多表分组连接查询的性能，实验过程中去掉了查询语句中的 order by 及 having 等部分。采用 TPC-H 中的数据生成器产生数据，并根据扩展因子产生不同大小的数据集 5 GB、10 GB、15 GB、20 GB。

5.2　性能测试

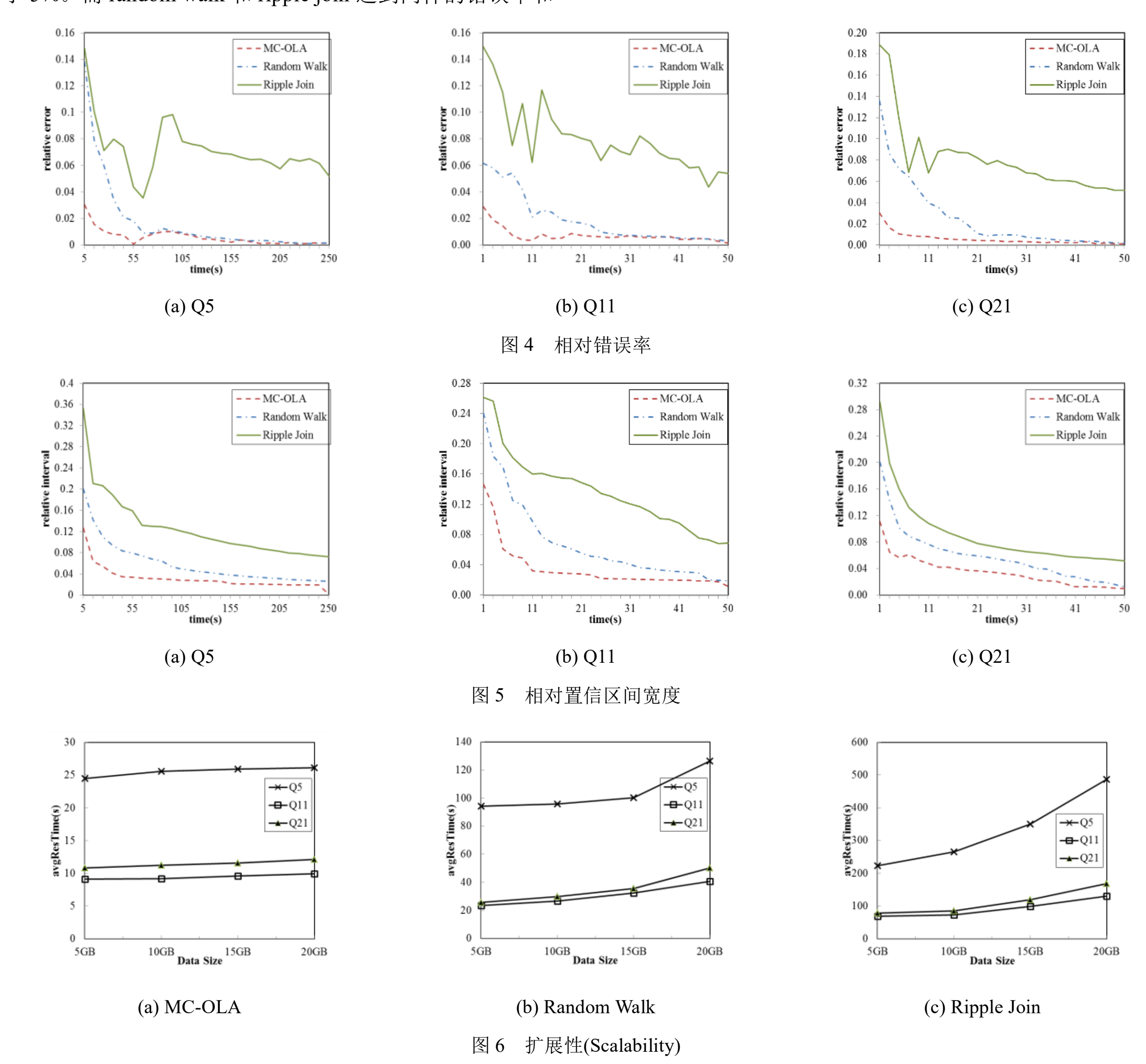
实验过程中，将置信度设置为 95%，聚集结果更新间隔设置为 2%，即每当查询进程增加 2%时更新一次聚集结果。采用相对错误率 *relative_error* 衡量估计结果的准确性，在分组聚集查询中，计算每个分组估计结果的相对错误率，并对所有的数据进行平均来衡量整个查询语句的准确性，其计算方式为

$$relative_error = \frac{1}{N}\sum_{i=1}^{N}\frac{|estimateValue_i - actualValue_i|}{actualValue_i}$$

采用相对置信区间宽度 *relative_interval* 衡量置信区间的收敛程度，与相对错误率类似，计算各个分组的相对置信区间宽度的平均值，其计算公式为

$$relative_interval = \frac{1}{N}\sum_{i=1}^{N}\frac{half\ \ width\ of\ \ confidence\ interval}{estimate\ result}$$

为尽量减少实验偏差的影响，所有负载执行三次，取其平均值进行结果分析和展示。图 4 和 5 分别展示了在 10GB 大小的数据集上三个查询的平均相对错误率和相对置信区间宽度。MC-OLA 在准确性和估计结果收敛速度方面均明显优于 random walk 和 ripple join。以六表连接查询 Q5 为例，MC-OLA 在 25s 左右时相对错误率已经小于 1%，相对置信区间宽度已经小于 5%。而 random walk 和 ripple join 达到同样的错误率和置信区间宽度的时间分别是 95s 和 265s 左右。在查询 Q11 和 Q21 上，MC-OLA 的响应速度也有类似的表现，优于 random walk 3 倍左右，优于 ripple join 一个数量级。一方面，MC-OLA 和 random walk 均利用连接列上的索引进行随机游走实现多表连接，ripple join 分别从多个表中进行统一随机抽样进行连接，而且样本之间并不是独立的，所以 MC-OLA 和 random walk 的执行复杂度更低，连接“目的性”更强，连接结果的产生率要高于 ripple join。另一方面，random walk 和 ripple join 分别从连接起始表和所有连接表中进行统一随机抽样，而 MC-OLA 采用分层采样技术，并且将每个分组的连接结果作为总体分别进行估计，因此在查询处理前期即可获得小分组的数据，从而提高了小分组的估计准确性和收敛速度。



(a) Q5　　(b) Q11　　(c) Q21

图 4　相对错误率

(a) Q5　　(b) Q11　　(c) Q21

图 5　相对置信区间宽度

(a) MC-OLA　　(b) Random Walk　　(c) Ripple Join

图 6　扩展性(Scalability)

5.3　扩展性测试

扩展性测试通过改变数据集大小实现，采用平均响应时间 *avgResTime* 衡量响应速度，*avgResTime* 表示估计结果的相对错误率开始小于 1%并且相对置信区间宽度开始小于 5%时的时间。图 6 展示了三种算法在数据集大小为 5 GB、10 GB、15 GB 和 20 GB 上的平均响应时间。可以看出，随着数据集的增大，MC-OLA 的平均响应时间并无太大变化，random walk 和 ripple join 的平均响应时间则随着数据集的增大而缓慢增长。MC-OLA 从样本起始层抽取样本后，利用连接列上的索引从相邻元组中抽样进行下一步游走。经过 *n* 次游走后得到的连接结果数只与马

尔可夫链上节点的出度、游走的成功率以及游走次数 n 有关，与原始数据集的规模无关，因此平均响应时间理论上与数据规模无关。尽管 random walk 也是通过随机游走实现连接，然而在分组连接查询处理中，小分组的估计相对需要更多的样本数据，且随着数据集的增大，抽取到小分组样本的概率有所降低，因此平均响应时间会随着数据集的增大而有所增加。Ripple join 随机从 k 张连接表中抽取 n 个样本，设各连接表之间元组的连接度数为 d，每个表大小为 N，则能得到的连接结果数为 $n^k(\frac{d}{N})^{k-1}$，因此平均响应时间呈现出低于数据集线性增长的趋势。

6 结束语

多表连接查询是大数据分析领域最重要的操作之一，然而多表连接的实现复杂度较高，导致其运行时间较长，从而影响了数据分析任务的时效性和价值。在线聚集能够在查询完成之前反馈具有统计意义的估计结果和置信区间，因而提供了一种有效减少运行时间和计算代价的查询实现方式。目前已有的多表连接在线聚集算法以 ripple join 为主，从各个连接表中进行统一随机采样，分别实现新数据和旧数据的连接，然后进行结果估计。这种算法的复杂度较高，且当连接选择率较低时导致结果产出率极低。除此之外，对于分组连接查询，还会导致小分组的估计准确度较低。本文提出了基于马尔可夫链的多表连接在线聚集技术 MC-OLA，其主要贡献包括：提出了实现多表连接在线聚集的系统架构；使用马尔可夫链对多表连接进行建模，将连接过程转换为马尔可夫链上的随机游走过程；提出了多表连接的顺序确定方法和游走起始层的分层采样算法，并提出了相应的采样策略和结果估计算法。在 HOP 平台上的实验结果表明，MC-OLA 的平均响应时间优于 random walk 以及 ripple join，而且具有良好的数据扩展性。目前 MC-OLA 只支持多表连接分组查询，在今后的工作中，将围绕嵌套查询、多查询的在线优化等展开进一步的研究。

参考文献：

[1] TPC-H Bechmark [EB/OL]. [2018-07-20]. http://www. tpc. org/tpch/.

[2] Condie T, Conway N, Alvaro P, *et al.* Online aggregation and continuous query support in mapreduce [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2010: 1115-1118.

[3] Hellerstein J, Haas P, Wang H. Online aggregation [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 1997: 171-182.

[4] Haas P. Large-Sample and deterministic confidence intervals for online aggregation [C]// Proc of the 9th Conference on Scientific and Statistical Database Management. 1997: 51-63.

[5] Pansare N, Borkar V, Jermaine C, *et al.* Online aggregation for large mapreduce jobs [J]. Proceedings of the VLDB Endowment, 2011, 4 (11): 1135-1145.

[6] Kai Z, Sameer A, Ankur D, *et al.* G-OLA: generalized on-line aggregation for interactive analysis on big data [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2015: 913-918.

[7] Sameer A, Barzan M, Aurojit P, *et al.* BlinkDB: queries with bounded errors and bounded response times on very large data [C]// Proc of the 8th Eurosys Conference. New York: ACM Press, 2013: 29-42.

[8] Qin C, Florin R. PF-OLA: a high-performance framework for parallel online aggregation [J]. Distributed and Parallel Databases, 2014, 32 (3): 337-375.

[9] Cheng Yu, Zhao Weijie, Florin R. Bi-Level Online Aggregation on Raw Data [C]// Proc of the 29th International Conference on Scientific and Statistical Database Management. New York: ACM Press, 2017: 10.

[10] Zhang Anzhen, Li Jianzhong, Gao Hong, *et al.* CrowdOLA: online aggregation on duplicate data powered by crowdsourcing [J]. Journal of Computer Science and Technology, 2018, 33 (2): 366-379.

[11] Haas P, Hellerstein J. Ripple joins for online aggregation [C]// Proc of International Conference on Management of Data. New York: ACM Press, 1999: 287-298.

[12] Luo G, Curt J, Peter J, *et al.* A scalable hash ripple join algorithm [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2003: 252-262.

[13] Chris J, Subramanian A, Abhijit P, *et al.* Scalable approximate query processing with the DBO engine [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2007: 725-736.

[14] Chris J, Subramanian A, Abhijit P, *et al.* Scalable approximate query processing with the DBO engine [J]. ACM Trans on Database Systems, 2008, 33 (4): 23.

[15] Shi Yingjie, Meng Xiaofeng, Wang Fusheng, *et al.* You can stop early with COLA: Online processing of aggregate queries in the cloud [C]// Proc of the 21st ACM Conference on Information and Knowledge Management. New York: ACM Press, 2012: 1223-1232.

[16] Li Feifei, Wu Bin, Yi Ke, *et al.* Wander join and XDB: online aggregation via random walks [J]. SIGMOD Record, 2017, 46 (1): 33-40.

[17] 史英杰, 杜方, 尤亚东. MSOLA: 基于多维分层采样的大数据在线聚集技术 [J]. 计算机应用研究, 2018, 35 (2): 61-66. (Shi Yingjie, Du Fang, You Yadong. MSOLA: big data online aggregation based on multi-dimension stratified sampling [J]. Application Research of Computers, 2018, 35 (2): 61-66.)

[18] Swarup A, Phillip B, Viswanath P. Congressional samples for approximate answering of group-by queries [C]// Proc of ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2000: 487-498.

[19] Horvitz D, Thompson, D. A generalization of sampling without replacement from a finite universe [J]. Journal of the American Statistical Association, 1952, 47 (260): 663–685.